Medical Blockchains: First Example Bitcoin Mechanics I

Alexander Schönhuth

UNIVERSITÄT BIELEFELD
Faculty of Technology

Bielefeld University
June 1, 2022

Recap Lecture 5

- *Incentives*
  - Block Rewards
  - Transaction Fees
- *Proof of Work*
  - Sybil attack: Recap
  - Proof of work: Key idea
  - Hash puzzles
- *Hash Puzzles: Properties*
  - Difficult to compute
  - Parameterizable cost
  - Trivial to verify
- *Mining Cost, Bootstrapping, 51-Percent Attack*
  - Mining cost: Basic calculation & complications
  - Bootstrapping: Feedback loop & recruiting miners
  - 51-Percent-Attack: Issues to consider

Medical Blockchain Motivation

Medical Blockchain Overview

Medical Blockchain Elements

Bitcoin Mechanics I

Overview

- *Medical Blockchain: Motivation*
  - Situation, risks, goals
  - Attribute Based Encryption
  - Key Aggregate Cryptography
  - Cloud based solutions
- *Medical Blockchain: Overview*
  - Nodes and data
  - Access rights
  - Transactions
  - Block structure
- *Medical Blockchain: Elements*
  - Transaction types: details
  - Tokens & rewards
  - Election
- *Bitcoin Mechanics I*
  - Transactions in detail
  - Metadata, Input, Output

Medical Blockchain Motivation

Medical Blockchain Overview

Medical Blockchain Elements

Bitcoin Mechanics I

Motivation I

Situation

- Medical data scattered across institutions
  ☞ Strict regulations prevent sharing and transferring
- Data standards vary per institution
  ☞ Low level of interoperability
- Little people authorized to provide access and distribute
  ☞ Patients have trouble to use their data

☞ Optimal usage of *individual medical histories* hardly possible

Motivation II

Risks

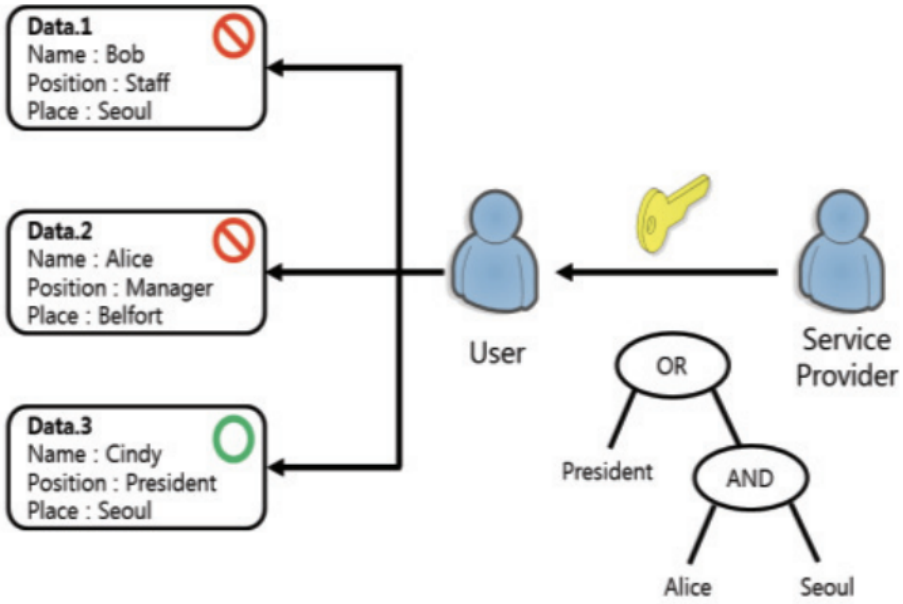
- No guarantee on reliability and integrity of patient data
  - Loss or hacking are real possibilities
  - Personal privacy leaks, data security, etc.
- Medical data stored in centralized manner
  - Allows malicious tampering, prone to hacking
  - Node failure due to natural disasters
- *Examples:*
  - *June 2017, Bondi Junction, Australia:* Accidental release of identifiable patient records
  - *October 2017:* 47 GB medical records stored in Amazon database accidentally opened to public

Motivation III

Goals

- Enable seamless exchange and sharing of medical data
- Get rid of single points of failure
- Make data tamper-proof and resistant to attacks
- Make data verifiable and immutable
- *Cloud based solutions rely on*
  - *Attribute-based encryption* schemes
  - *Key aggregate cryptography* based safety

Attribute Based Encryption I

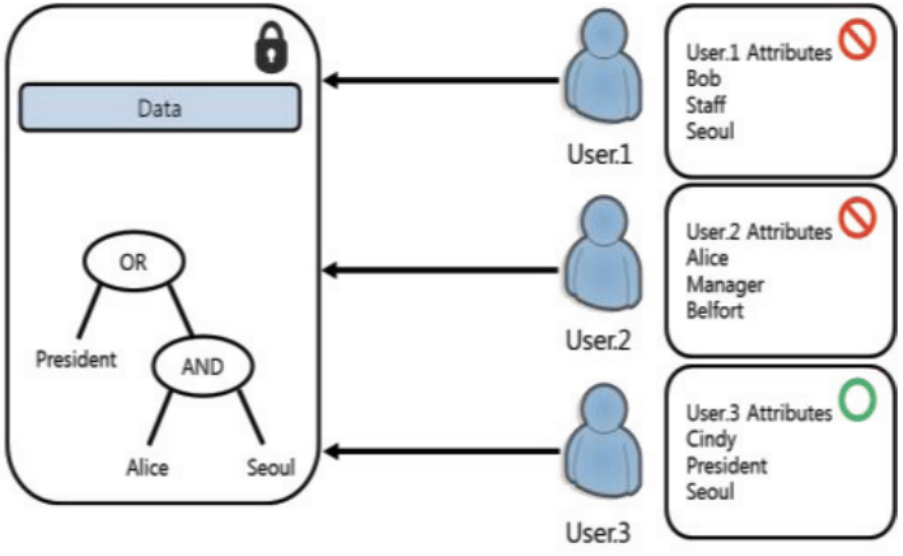


Key Policy Attribute Based Cryptography

From [Lee et al., 2015]

- Keys for decrypting data depend on *user properties*
- User characteristics part of key ☞*"key based"*
- Service provider encrypts data and issues keys on demand

Attribute Based Encryption II

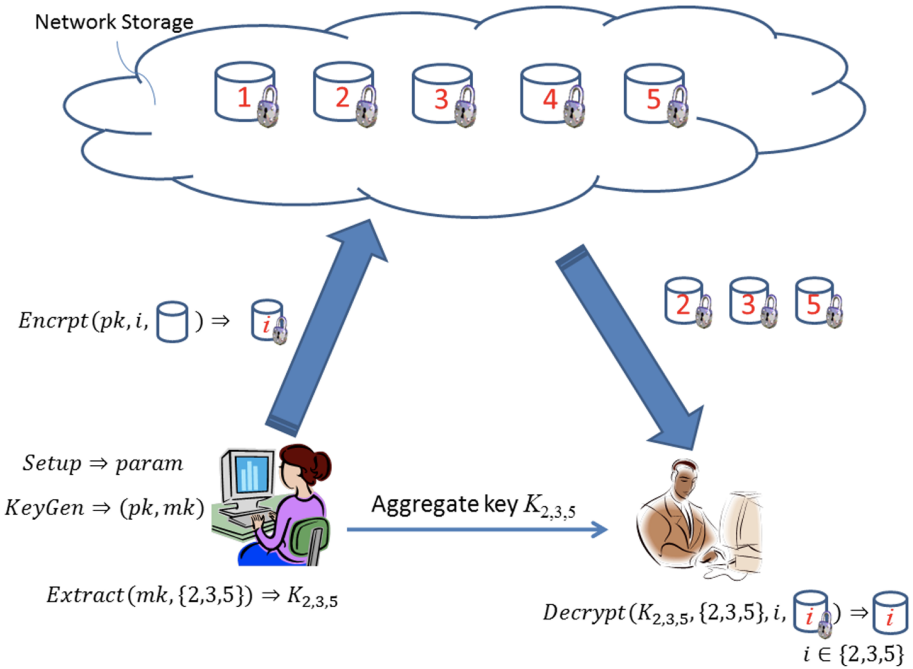


Cyphertext Policy Attribute Based Cryptography

From [Lee et al., 2015]

- Keys for decrypting data depend on *data characteristics*
- Data encrypted involving characteristics
  ☞ *"cyphertext based"*
- Service provider encrypts and issues particularly tailored keys

Key Aggregate Cryptography I

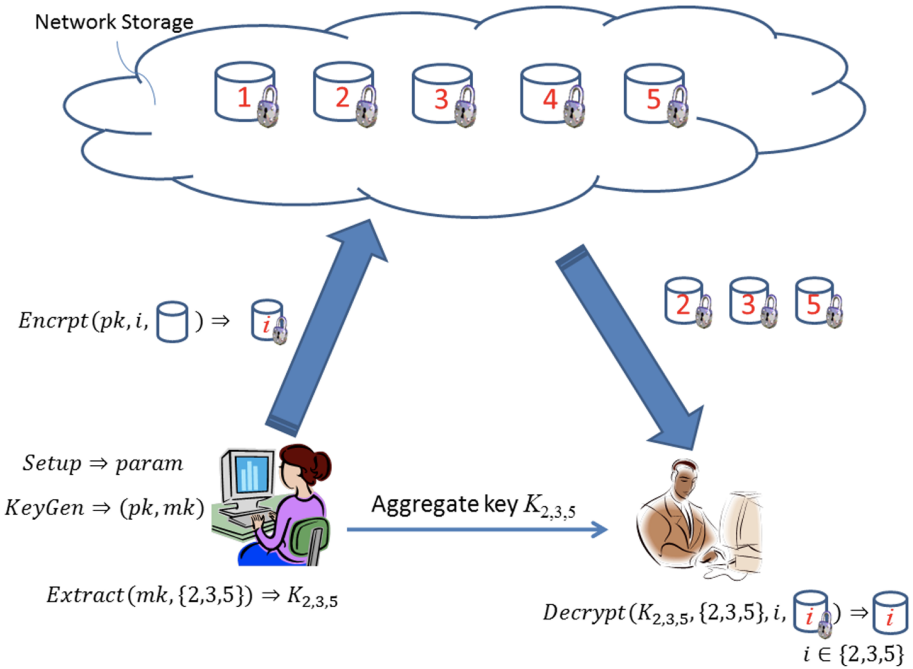


Key Aggregate Cryptography

From [Chu et al., 2014]

- *Master* generates key pair (MK, PK)
  - MK is *master key*
  - PK is *public key*
- *Master* sets all necessary parameters
- *Master* encrypts each data record i using PK

Key Aggregate Cryptography II



Key Aggregate Cryptography

From [Chu et al., 2014]

- *User* requests access for data records $i_1, \ldots, i_l$
- *Master* generates *aggregate key* $K_{i_1,\ldots,i_l}$ using MK
- *Master* sends $K_{i_1,\ldots,i_l}$ to *User*
- *User* decrypts data records $i_1, \ldots, i_l$ using $K_{i_1,\ldots,i_l}$

Cloud Based Solutions: Drawback

- ABE and KAC alone provide
  - Tamper resistance
  - Privacy protection
  - Secure storage
- ABE and KAC alone do *not* provide
  - Full control of individual medical records by patient
  - Independence from third parties
    ☞ Cloud management holds master keys etc.
- *Solution:* Integrate cloud based with blockchain based system.
- *Consequences:*
  - Patients ”own” their data
  - Cloud management decentralized

Medical Blockchain Motivation

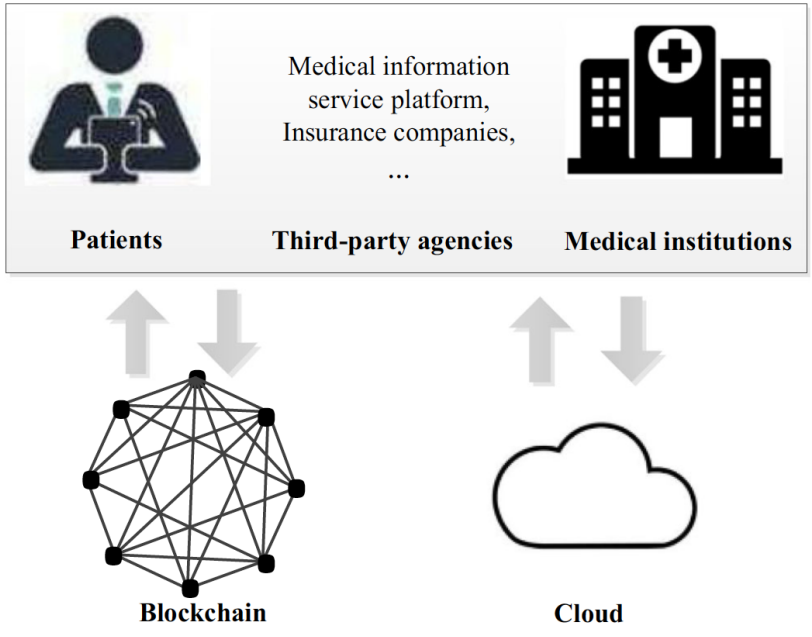
Medical Blockchain Overview

Medical Blockchain Elements

Bitcoin Mechanics I

Medical Blockchain: Overview

- Data too large ☞ needs to be put to cloud
- Different participants have different ...
  - ... requirements
  - ... rights to access data
- *Challenge:* Design appropriate blockchain
  - Cloud holds data
  - Blockchain holds metadata

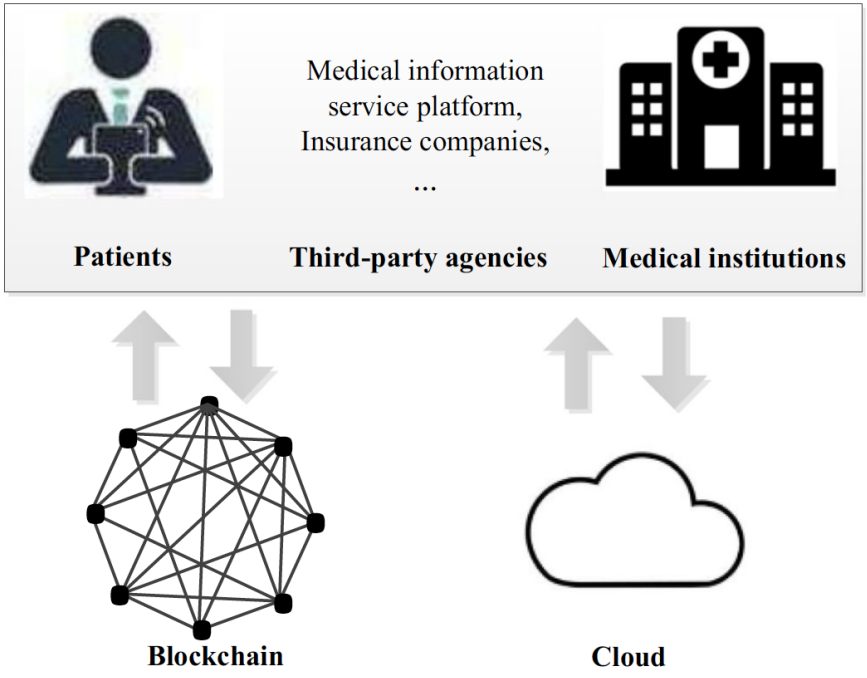


Medical Blockchain Architecture

From [Chen et al., 2019]

Medical Blockchain: Nodes and Data

- Patients
- Third-party agencies
  - Research teams
  - Insurance companies
  - Information platforms
- Medical Institutions
- *Data:* Individual medical records

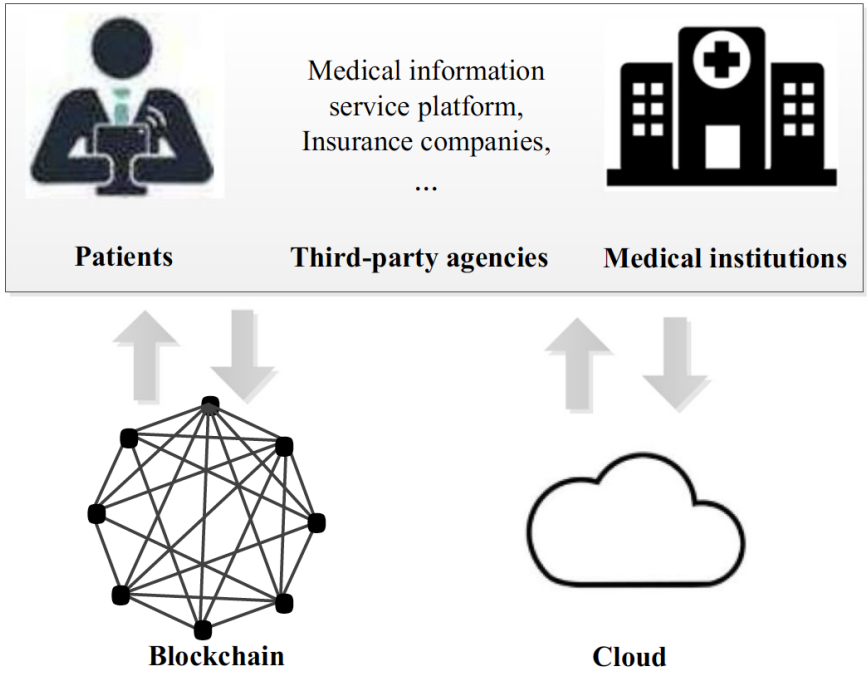


Medical Blockchain Architecture

From [Chen et al., 2019]

Medical Blockchain: Access Rights

- Everyone has read/write access to own data
- By default, everyone gets access to other data only by consent of data owner
- *Exception – Emergency:* Medical institutions get read access without authorization

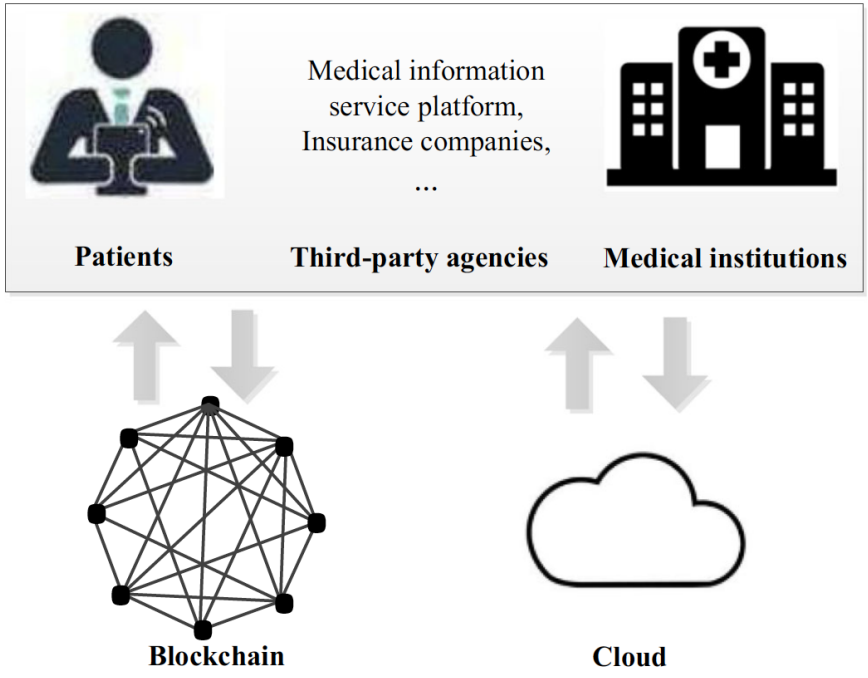


Medical Blockchain Architecture

From [Chen et al., 2019]

Medical Blockchain: Transactions

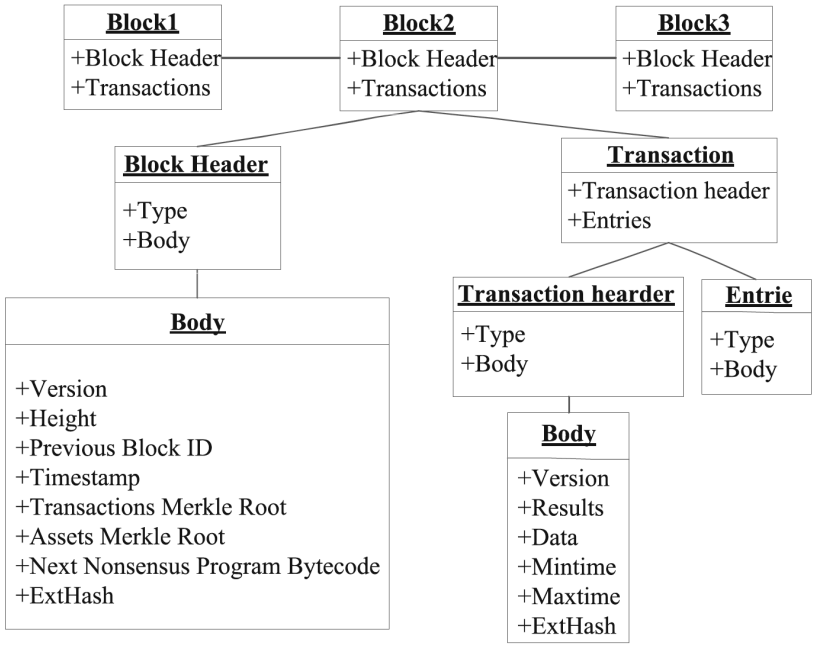
- Medical data *release*
  ☞ through medical institution / doctor
- Medical data *storage*
  ☞ through patient
- Medical data *sharing*
  ☞ patient authorizes third-party or external institution



Medical Blockchain Architecture

From [Chen et al., 2019]

Medical Blockchain: Block Structure



Medical Blockchain: Block Structure

From [Chen et al., 2019]

- Transactions arranged by Merkle tree
- Transactions contain hash of medical data in cloud
- Public key encryption is used for data in cloud
- Access control policy established by cloud storage management

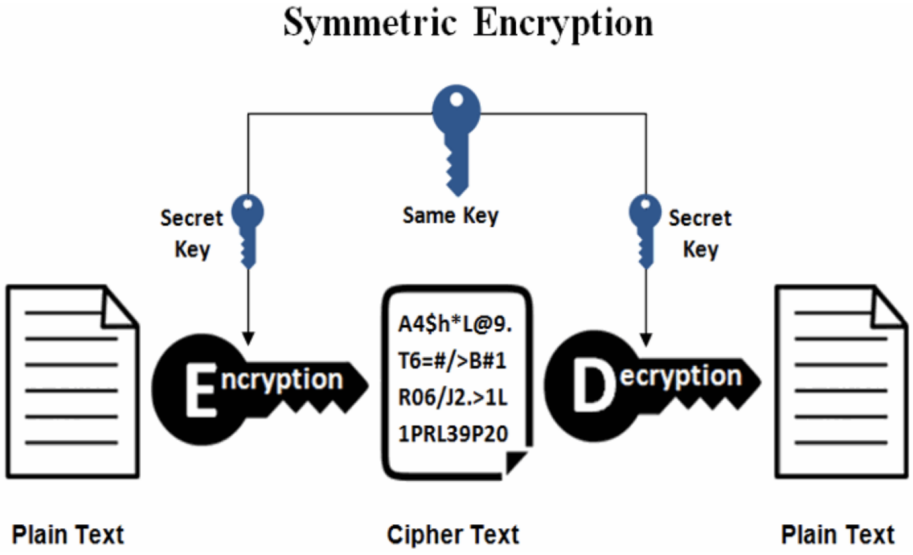
Medical Blockchain Motivation

Medical Blockchain Overview

Medical Blockchain Elements

Bitcoin Mechanics I

Symmetric Encryption

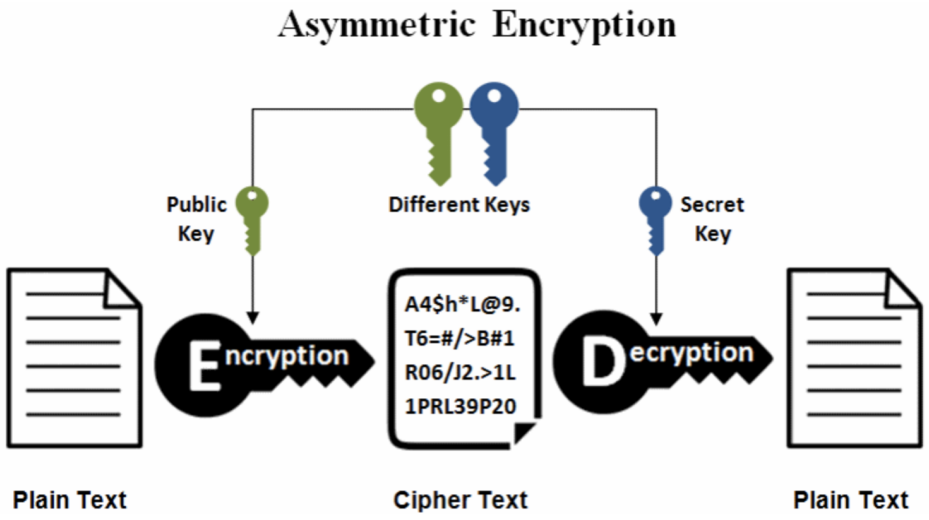


Symmetric Key Encryption Scheme

From `wizardforcel.gitbooks.io/`

- Encrypting and decrypting party share identical key
- *Advantage:* Data securely shared between two parties
- *Disadvantage:* Keys need to be transmitted across network

Asymmetric Encryption I

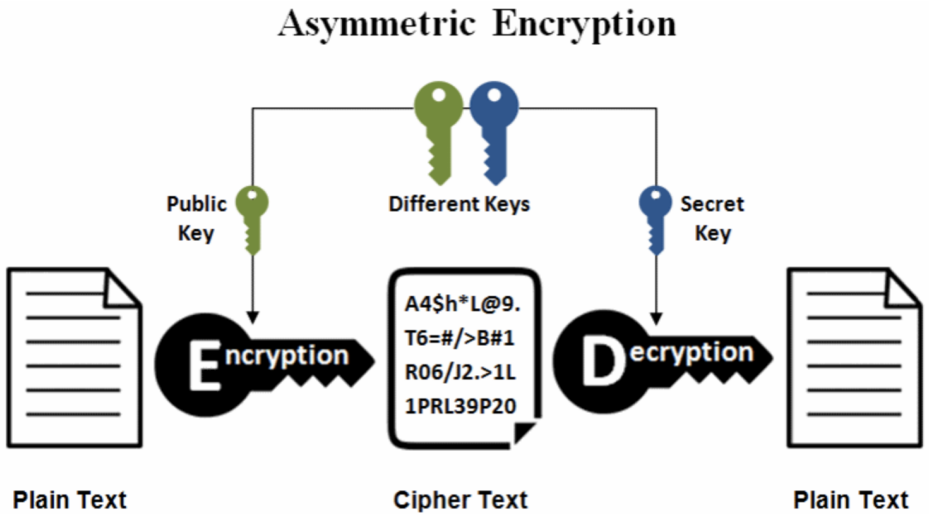


Asymmetric Key Encryption Scheme

From `wizardforcel.gitbooks.io/`

- *Encryption:* Everyone holding public key can encrypt data
- *Decryption:* Only secret key holder can decrypt
- *Advantage:* No transmission of secret keys necessary

Asymmetric Encryption II



Asymmetric Key Encryption Scheme

From `wizardforcel.gitbooks.io/`

- *Safe transmission of secret key S from A to B:*
  - B generates asymmetric key pair (PK, SK), shares PK with A
  - A encrypts S with PK
  - B decrypts S with SK
- *Example:* S being symmetric, attribute based, or aggregate key

Medical Blockchain Transactions: Release

Medical Data Release: Situation

- Patient A visits doctor / medical institution B
- A and B have key pairs (S_A, P_A) and (S_B, P_B)
- *Reminder:* P_A and P_B are A's and B's blockchain identities
- B generates data D_A and performs transaction

Medical Data Release: Operations

- B generates digest $H(D_A)$ using hash function H
- B signs $H(D_A)$ using his secret key S_B
- *Transaction:* Records A, B, and $H(D_A)$ with signature from B

Medical Blockchain Transaction: Storage I

Medical Data Storage: Situation

- Patient A with (S_A, P_A) has visited doctor / medical institution B with (S_B, P_B), who generated data D_A
- B has released data generation record to blockchain
- *Simultaneously:*
  - B generates symmetric key K_S
  - B encrypts D_A with K_S and K_S with P_A, yielding D'_A and K'_S
  - B sends D'_A and K'_S to A
- A is to perform storage transaction

Medical Blockchain Transaction: Storage II

Medical Data Storage: Operations

- A verifies signature of B on $H(D_A)$ on blockchain
- A decrypts K'_S using S_A and D'_A using K_S
- A generates new encryption key E_A for cloud storage (e.g. a master-public key pair for KAC)
- *Cloud:*
  - A encrypts data D_A using E_A, yielding D''_A
  - A signs D''_A using S_A
  - A stores both D''_A and its signature
- *Blockchain:*
  - Hash pointer to release transaction in blockchain
  - Hash pointer to D''_A in cloud
  - Transaction signed using S_A

Medical Blockchain Transaction: Sharing

Medical Data Sharing: Situation

- Third party C with (S_C, P_C) requests access to data $D_{A1}, ..., D_{Al}$ stored on cloud from A
- Release and storage of all $D_{A1}, ..., D_{Al}$ recorded in blockchain

Medical Data Sharing: Operations

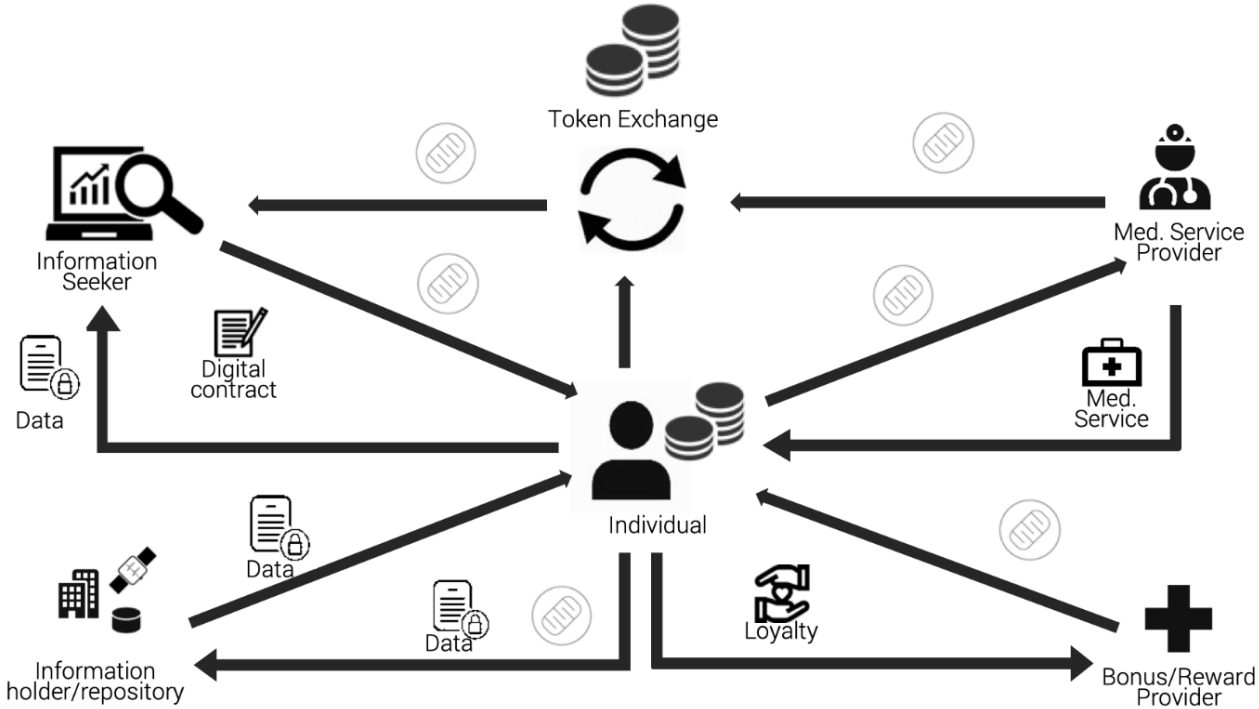
- *Cloud: A is the authority for self-owned data*
  - A generates decryption key(s) for $D_{A1}, ..., D_{Al}$
  - A sends key(s) to C, who accesses and decrypts $D_{A1}, ..., D_{Al}$
- *Blockchain:*
  - Access for P_C signed by S_A
  - Proof of access: deposit decryption key(s) (and additional rights)
  - Hash pointers to storage transactions of $D_{A1}, ..., D_{Al}$

Medical Blockchain: Tokens I

Currency and Tokens

- Tokens are digital assets of value
- For a cryptocurrency:
  - Coins themselves
  - Rights to purchase something
- Medical data is valuable:
  - The individual itself requires it for optimizing health
  - Health insurances require it for optimizing premiums
  - Research teams require it for drawing scientificaly competitive conclusions

Medical Blockchain: Tokens II



Token Economy

From `tectales.com`

- Medical data raises a market in its own right
- *Idea:* Reward miners with tokens of interest

Medical Blockchain: Rewards

- *Transaction fees*
  - Requires platforms for connecting to external services
  - *Note:* Such platforms are referred to as *oracles*
- *Block reward patients:*
  - Treatment vouchers
  - Health insurance premium reductions
- *Block reward hospitals:*
  - Receive funds for / access to external medical data
  - Raises competitiveness in performing studies
  - May optimize in-house protocols
- *Block reward health insurances:*
  - Access to patient data

Medical Blockchain: Election

Delegated Proof of Stake

- *Currency:* Tokens
- *Proof of Stake:* Chances proportional to medical data owned
- *Delegated Proof of Stake:*
  - Assign your tokens to delegates via *staking pool*
  - Raises chances of delegates
  - No participation by oneself required
  - Long-term trusted delegates attract others' tokens

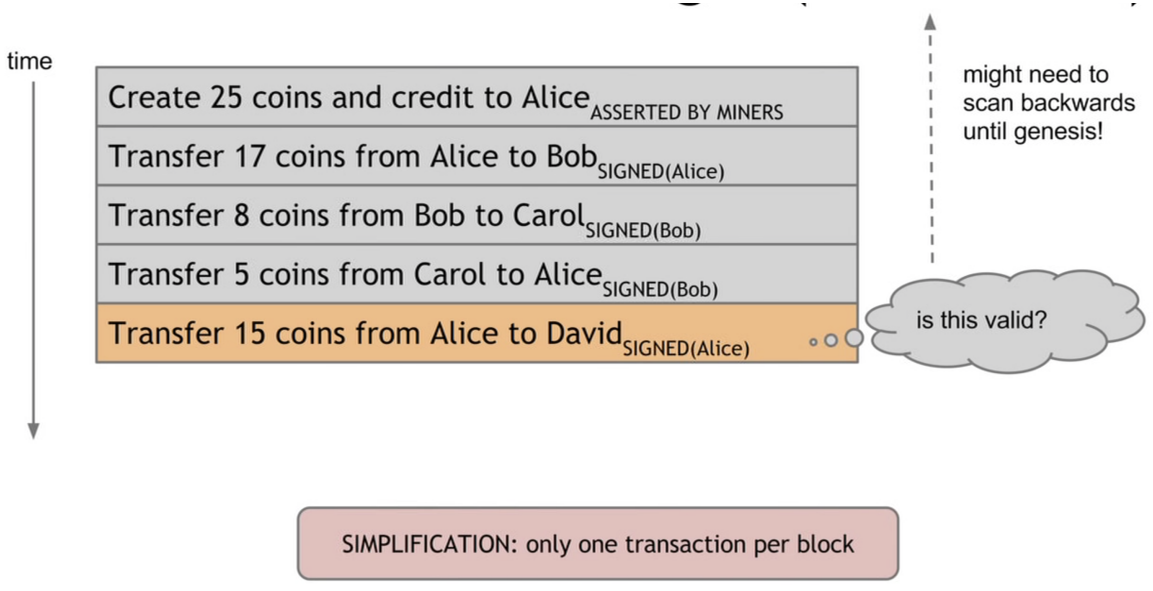
Medical Blockchain Motivation

Medical Blockchain Overview

Medical Blockchain Elements

Bitcoin Mechanics I

Account Based Ledger I

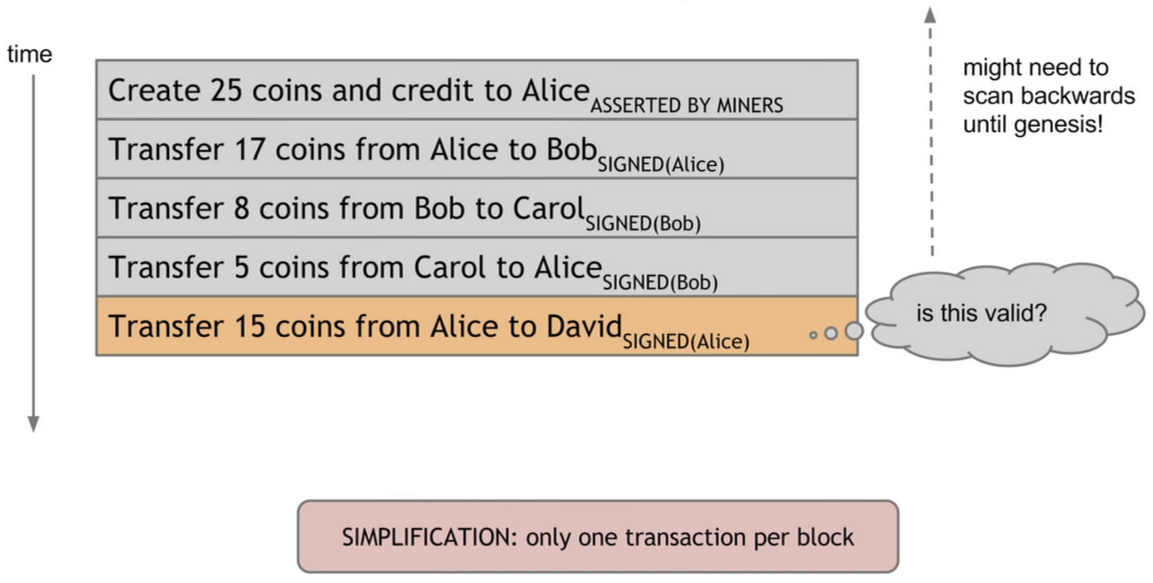


Account based ledger

From `bitcoinbook.cs.princeton.edu`

- Transactions recorded in terms of sender and recipient
- Requires to keep track of identities
- Requires to keep track of identities' accounts

Account Based Ledger II

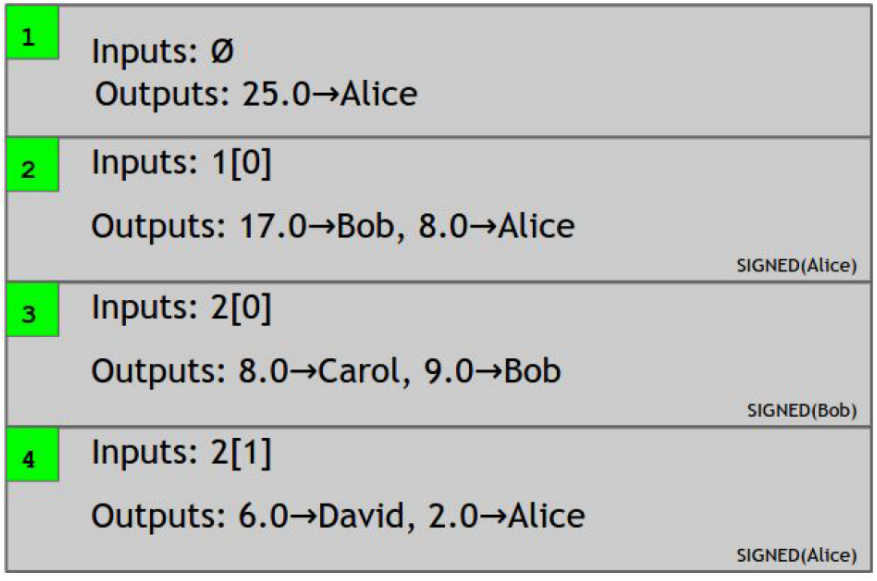


Account based ledger

From `bitcoinbook.cs.princeton.edu`

- Verifying transactions requires to keep track of accounts
- Requires to look back in history
- Expensive operation
  - Additional data structures increase efficiency...
  - ... but require housekeeping beyond the blockchain

Transaction Based Ledger I



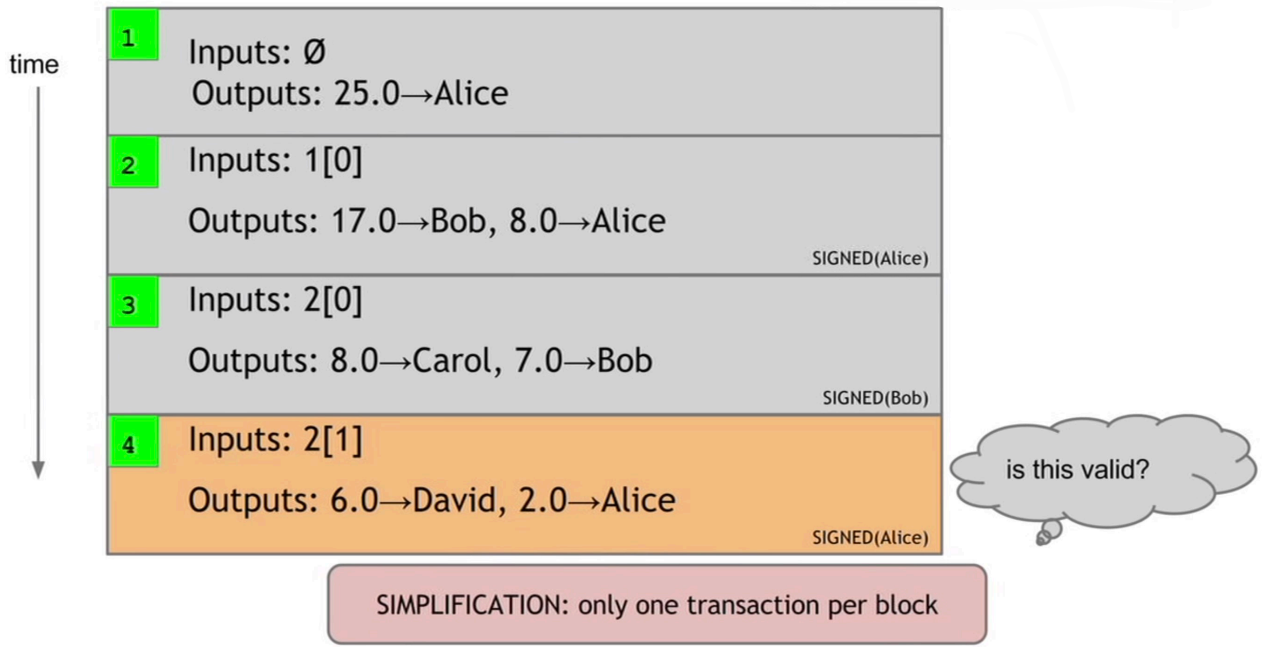
Transaction Based Ledger

Note: one transaction per block

From `bitcoinbook.cs.princeton.edu`

- *Reminder:* Do similar as for ScroogeCoin
- Each transaction input indexed as $x[y]$
  - y indexes transaction
  - x indexes output
- Each transaction signed by input provider

Transaction Based Ledger II

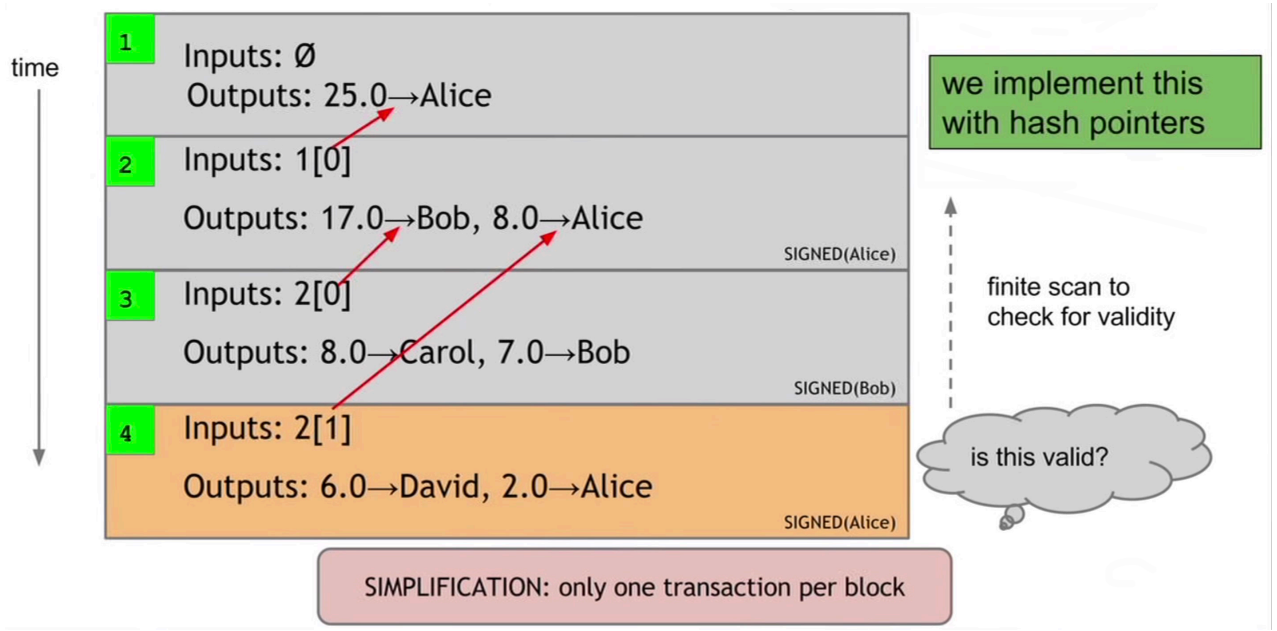


Transaction Based Ledger: Validity Check

From `bitcoinbook.cs.princeton.edu`

How to check validity? Any gains in processing time?

Transaction Based Ledger III



Transaction Based Ledger: Finite Scan

From `bitcoinbook.cs.princeton.edu`

Checking validity amounts to finite scan

Transaction Based Ledger: Features

Change Addresses

- The output needs to consume the input
- Remaining input needs to be routed to a *change address*
- Change address could be input address or alternative identity

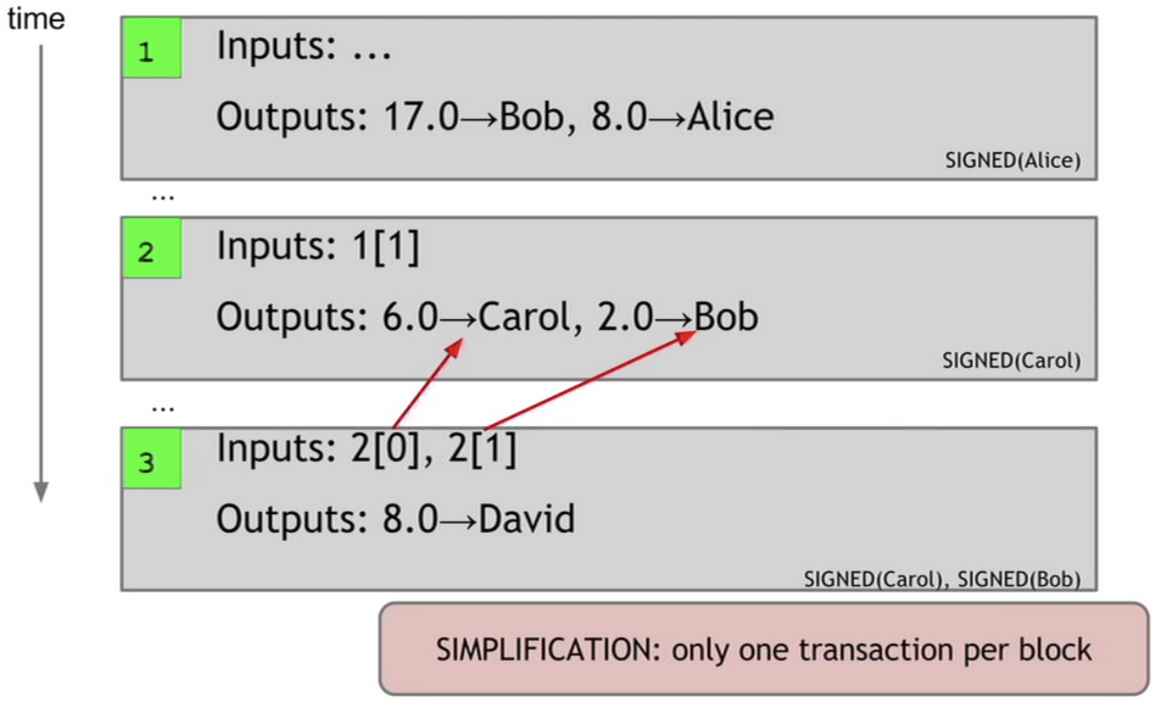
Efficient Verification

- Compare with transactions whose output is consumed
  ☞ Constant time operation

Consolidating Funds

- *Goal:* Summarize different outputs owned by same node
- Create transaction that
  - collects different outputs as its input
  - has one output sent one of node's identities

Transaction Based Ledger: Features II



Joint Payments

From `bitcoinbook.cs.princeton.edu`

- Transaction claiming outputs from several identities as input
- All identities providing input sign transaction

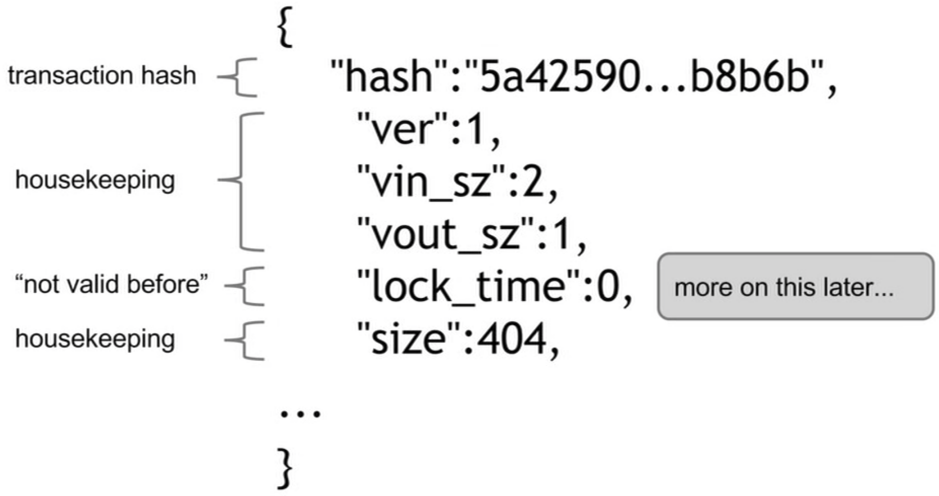
Transaction Syntax

Transaction Syntax

From `bitcoinbook.cs.princeton.edu`

- Each transaction virtually is string of bits
- Scripting language supports compilation of transactions
- Parts of script: 1.) metadata, 2.) inputs, 3.) outputs

Transaction Syntax: Metadata



Transaction Metadata Syntax

From `bitcoinbook.cs.princeton.edu`

- Housekeeping information: size, number of inputs and outputs
- Hash of transactions ☞ unique ID for transaction
- ”lock_time” field: to be discussed later

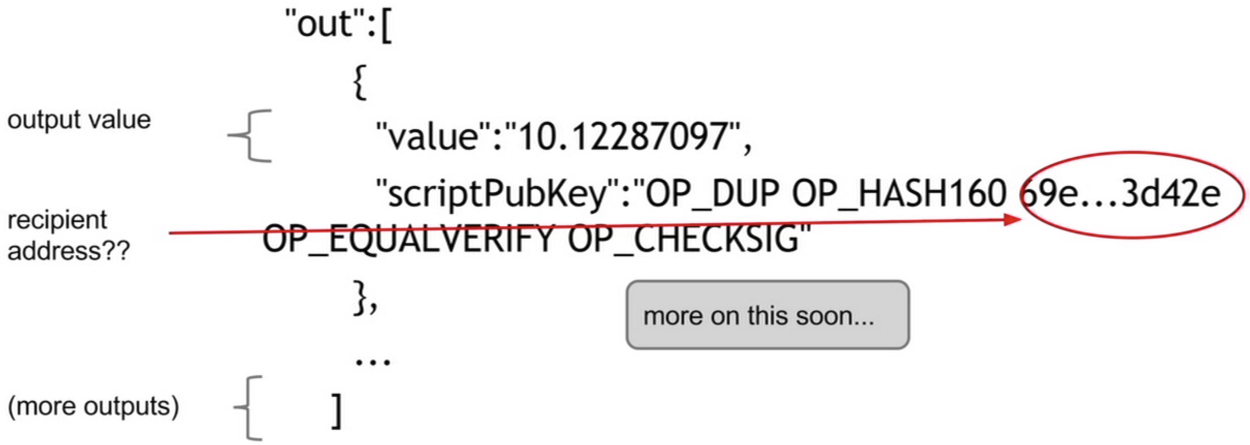
Transaction Syntax: Inputs

Transaction Input Syntax

From `bitcoinbook.cs.princeton.edu`

- Inputs form an array; each input has same form
- Hash pointer to previous transaction ☞ previous output is input
- Signature of the owning identity (identities)

Transaction Syntax: Outputs



Transaction Output Syntax

From `bitcoinbook.cs.princeton.edu`

- Outputs form an array: each output has two fields
- Value where sum of output values at most sum of input values ☞ difference: transaction fee
- Field specifying recipient(s); in fact, that field is a *script*

Materials / Outlook

- See [Chen et al., Journal of Medical Systems 43(5), 2019]; `https://doi.org/10.1007/s10916-018-1121-4`
- See *Bitcoin and Cryptocurrency Technologies*, 3.1
- See `https://bitcoinbook.cs.princeton.edu/` for further resources
- See also related resources at `https://www.tectales.com`
- Next lecture: “Bitcoin Mechanics II: Scripts”
  - See *Bitcoin and Cryptocurrency Technologies* 3.2 & 3.3
- ”Smart contracts and Ethereum I”
  - See *Bitcoin and Cryptocurrency Technologies* 10.7